EXHIBIT 11

UNITED STATES PATENT NO. 11,805,267

CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

As demonstrated in the chart below, ASUS directly and indirectly infringes the asserted claims of US 11,805,267 (the "'267 Patent"). ASUS directly infringes, contributes to the infringement of, and/or induces infringement of the '267 Patent by making, using, selling, offering for sale, and/or importing into the United States the Accused Products that are covered by one or more claims of the '267 Patent. The Accused Products are devices that decode AV1-compliant video and/or encode video into an AV1-compliant format. For example, ASUS Q543MV Notebook ("ASUS Q543MV") is a representative product for other ASUS devices that decode AV1-compliant video and/or encode video into an AV1-compliant format.

The ASUS Q543MV contains at least one video decoder that helps decode AV1-compliant video. Additionally, the ASUS Q543MV contains at least one video encoder that helps encode video into an AV1-compliant video format.[1] While evidence from the ASUS Q543MV is specifically charted herein, the evidence and contentions charted herein apply equally to the other ASUS Accused Products that decode AV1-compliant video. On information and belief, the evidence and contentions charted herein apply equally to the other ASUS Accused Products that encode video into an AV1-compliant format.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the '267 Patent. Moreover, this exemplary chart does not limit, and is not intended to limit, Nokia's infringement positions or contentions.

The following infringement chart also includes exemplary citations to the AV1 Bitstream & Decoding Process Specification, rev'd Jan. 8, 2019 (the "AV1 Specification").[2] Any ASUS device that includes a decoder that practices the functionality in the AV1 Specification ("AV1 Decoder") practices the claims of the '267 Patent. Thus, the Accused Products each practice the AV1 Specification and are covered by claims of the '267 Patent.

Nokia contends each of the following limitations is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents. [3]

---

[1] See, e.g., https://www.asus.com/us/laptops/for-home/everyday-use/asus-vivobook-pro-15-oled-q543/techspec/; https://www.intel.com/content/www/us/en/products/sku/236849/intel-core-ultra-9-processor-185h-24m-cache-up-to-5-10-ghz/specifications.html; https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new.

[2] Available here: https://aomediacodec.github.io/av1-spec/av1-spec.pdf

[3] This claim chart is based on the information currently available to Nokia and is intended to be exemplary in nature. Nokia reserves all rights to update and elaborate its infringement positions, including as Nokia obtains additional information during discovery.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>ACCUSED PRODUCTS</th></tr>
<tr><td>19. [A] A method for decoding a block of pixels, the method comprising:</td><td>Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding a block of pixels.

For example, and without limitation, the Asus Q543MV uses hardware-accelerated video decoding and includes an NVIDIA GeForce RTX 4060 Laptop graphics processing unit (“GPU”) and an Intel Core Ultra 9 Processor 185H.

<table>
<tr><th></th><th>Q543MJ</th><th>Q543MV</th></tr>
<tr><td>Processor</td><td>Intel® Core™ Ultra 9 Processor 185H 2.3 GHz (24MB Cache, up to 5.1 GHz, 16 cores, 22 Threads); Intel® AI Boost NPU up to 11TOPS</td><td>Intel® Core™ Ultra 9 Processor 185H 2.3 GHz (24MB Cache, up to 5.1 GHz, 16 cores, 22 Threads); Intel® AI Boost NPU up to 11TOPS</td></tr>
<tr><td>Graphics</td><td>NVIDIA® GeForce RTX™ 3050 6GB Laptop GPU
6GB GDDR6
Intel® Arc™ Graphics</td><td>NVIDIA® GeForce RTX™ 4060 Laptop GPU (233 AI TOPs)
8GB GDDR6
Intel® Arc™ Graphics</td></tr>
</table>
Source: https://www.asus.com/us/laptops/for-home/everyday-use/asus-vivobook-pro-15-oled-q543/techspec/ (last accessed March 6, 2025).

<table>
<tr><td>H.264 Hardware Encode/Decode</td><td>Yes</td></tr>
<tr><td>H.265 (HEVC) Hardware Encode/Decode</td><td>Yes</td></tr>
<tr><td>AV1 Encode/Decode</td><td>Yes</td></tr>
</table>
</td></tr>
</table>

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	Source: https://www.intel.com/content/www/us/en/products/sku/236849/intel-core-ultra-9-processor-185h-24m-cache-up-to-5-10-ghz/specifications.html (last accessed March 6, 2025)(specifications for Intel Core Ultra 9 185H).

BOARD	FAMILY	NVENC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVENC	Max # of concurrent sessions	H.264 (AVCHD) YUV 4:2:0	H.264 (AVCHD) YUV 4:2:2	H.264 (AVCHD) YUV 4:4:4	H.264 (AVCHD) Lossless	H.265 (HEVC) 4K YUV 4:2:0	H.265 (HEVC) YUV 4:2:2	H.265 (HEVC) 4K YUV 4
GeForce RTX 4060 Laptop	Ada Lovelace	8th Gen	M	1	1	8	YES	NO	YES	YES	YES	NO	YES
GeForce RTX 4060	Ada Lovelace	8th Gen	D	1	1	8	YES	NO	YES	YES	YES	NO	YES

BOARD	FAMILY	NVDEC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVDEC	MPEG-1	MPEG-2	VC-1	VP8	VP9 4:2:0			H.264 (AVCHD) 4:2:0	
										8 Bit	10 Bit	12 Bit	8 Bit	10 Bit
GeForce RTX 4060 Laptop	Ada Lovelace	5th Gen	M	1	1	YES	YES	YES	YES	YES	YES	YES	YES	NO

H.265 (HEVC) 4:2:0			H.265 (HEVC) 4:2:2		H.265 (HEVC) 4:4:4			AV1	
8 Bit	10 Bit	12 Bit	8 Bit	10 Bit	8 Bit	10 Bit	12 Bit	8 Bit	10 Bit
YES	YES	YES	NO	NO	YES	YES	YES	YES	YES

Source: https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new (last accessed March 6, 2025)(row for 4060 Laptop GPU).

The AV1 decoder in the accused products practices a method for decoding a block of pixels, including when encoding a block of samples using the COMPOUND_REFERENCE prediction mode.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	# 1. Scope This document specifies the Alliance for Open Media AV1 bitstream formats and decoding process. AV1 Specification at p. 1 of 669. **Block** A square or rectangular region of samples. AV1 Specification at p. 1 of 669. **Compound prediction** A type of inter prediction where sample values are computed by blending together predictions from two reference frames (the frames blended can be the same or different). AV1 Specification at p. 2 of 669. **Encoder** One embodiment of the encoding process. **Encoding process** A process not specified in this Specification that generates the bitstream that conforms to the description provided in this document. AV1 Specification at p. 2 of 669. **Inter coding** Coding one block or frame using inter prediction. **Inter frame** A frame compressed by referencing previously decoded frames and which may use intra prediction or inter prediction. **Inter prediction** The process of deriving the prediction value for the current frame using previously decoded frames. AV1 Specification at p. 3 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
Prediction
The implementation of the prediction process consisting of either inter or intra prediction.

Prediction process
The process of estimating the decoded sample value or data element using a predictor.

Prediction value
The value, which is the combination of the previously decoded sample values or data elements, used in the decoding process of the next sample value or data element.
AV1 Specification at p. 4 of 669.

Sample
The basic elements that compose the frame.

Sample value
The value of a sample. This is an integer from 0 to 255 (inclusive) for 8-bit frames, from 0 to 1023 (inclusive) for 10-bit frames, and from 0 to 4095 (inclusive) for 12-bit frames.
AV1 Specification at p. 5 of 669.

<h3>6.10.24. Ref frames semantics</h3>
comp_mode specifies whether single or compound prediction is used:

<table>
<tr><th>comp_mode</th><th>Name of comp_mode</th></tr>
<tr><td>0</td><td>SINGLE_REFERENCE</td></tr>
</table>
</td></tr>
</table>

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
<table><tr><th>comp_mode</th><th>Name of comp_mode</th></tr><tr><td>1</td><td>COMPOUND_REFERENCE</td></tr></table>
SINGLE_REFERENCE indicates that the inter block uses only a single reference frame to generate motion compensated prediction.

COMPOUND_REFERENCE indicates that the inter block uses compound mode.
AV1 Specification at pp. 181-2 of 669.
</td></tr>
<tr><td>[B] determining, for a current block, a first reference block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;</td><td>
Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding video comprising determining, for a current block, a first reference block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision.

The Accused Products determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision when encoding a block of samples using the COMPOUND_REFERENCE prediction mode.

AV1 employs motion vectors for inter prediction processes. See AV1 Specification at §§ 7.10, 7.11.3.

Block
A square or rectangular region of samples.
AV1 Specification at p. 1 of 669.

Compound prediction
A type of inter prediction where sample values are computed by blending together predictions from two reference frames (the frames blended can be the same or different).
AV1 Specification at p. 2 of 669.
</td></tr>
</table>

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	(continued below)

Motion vector
A two-dimensional vector used for inter prediction which refers the current frame to the reference frame, the value of which provides the coordinate offsets from a location in the current frame to a location in the reference frame.

AV1 Specification at p. 4 of 669

Sample
The basic elements that compose the frame.

Sample value
The value of a sample. This is an integer from 0 to 255 (inclusive) for 8-bit frames, from 0 to 1023 (inclusive) for 10-bit frames, and from 0 to 4095 (inclusive) for 12-bit frames.

AV1 Specification at p. 5 of 669.

ref_frame_idx[i] specifies which reference frames are used by inter frames. It is a requirement of bitstream conformance that RefValid[ref_frame_idx[i]] is equal to 1, and that the selected reference frames match the current frame in bit depth, profile, chroma subsampling, and color space.

AV1 Specification at p. 153 of 669.

6.10.24. Ref frames semantics

comp_mode specifies whether single or compound prediction is used:

comp_mode	Name of comp_mode
0	SINGLE_REFERENCE

comp_mode	Name of comp_mode
1	COMPOUND_REFERENCE

SINGLE_REFERENCE indicates that the inter block uses only a single reference frame to generate motion compensated prediction.

COMPOUND_REFERENCE indicates that the inter block uses compound mode.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	AV1 Specification at pp. 181-2 of 669. 7.11.3. Inter prediction process 7.11.3.1. General The inter prediction process is invoked for inter coded blocks and interintra blocks. The inputs to this process are: • a variable plane specifying which plane is being predicted, • variables x and y specifying the location of the top left sample in the CurrFrame[plane] array of the region to be predicted, • variables w and h specifying the width and height of the region to be predicted, • variables candRow and candCol specifying the location (in units of 4x4 blocks) of the motion vector information to be used. AV1 Specification at pp. 257 of 669. 5. The variable refFrame is set equal to RefFrames[candRow][candCol][refList]. AV1 Specification at pp. 257 of 669. 8. The motion vector array mv is set equal to Mvs[candRow][candCol][refList]. AV1 Specification at pp. 258 of 669. 14. If isCompound is equal to 1, then the variable refList is set equal to 1 and steps 5 to 13 are repeated to form the prediction for the second reference. AV1 Specification at pp. 259 of 669. 7.11.3.4. Block inter prediction process The inputs to this process are:

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	• a variable plane, • a variable refIdx specifying which reference frame is being used (or -1 for intra block copy), • variables x and y giving the block location with in units of 1/1024 th of a sample, • variables xStep and yStep giving the step size in units of 1/1024 th of a sample, • variables w and h giving the width and height of the block in units of samples, • variables candRow and candCol specifying the location (in units of 4x4 blocks) of the motion vector information to be used. AV1 Specification at pp. 261-262 of 669. The sub-sample interpolation is effected via two one-dimensional convolutions. First a horizontal filter is used to build up a temporary array, and then this array is vertically filtered to obtain the final prediction. The fractional parts of the motion vectors determine the filtering process. If the fractional part is zero, then the filtering is equivalent to a straight sample copy. AV1 Specification at pp. 262 of 669.
[C] using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;	Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding video comprising using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision. Each of the Accused Products performs a method for decoding video comprising using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision. The following specifications provide further evidence of how each of the Accused Products operates. When the prediction is generated by interpolation, the procedure at section 7.11.3.4 is used. AV1 uses subpel_filter[][][], reproduced below, which uses a set of weights that total 128 (2^7), adding 7 bits of bit depth in both a horizontal and and vertical filtering step. After the first (horizontal) pass, the intermediate

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	result is rounded and right-shifted by InterRound0 (which equals 3, except when bit depth is 12, then is equal to 5). After the second horizontal pass, the intermediate result is rounded and right-shifted by InterRound1 (which is 7 when iscompund is equal to 1, except when bit depth is 12, and then is equal to 5). When iscompound is equal to 1 and useWarp is 0, then, AV1 generates a prediction with a precision 4 greater than the input bit depth. 13. If useWarp is equal to 0, the block inter prediction process in section 7.11.3.4 is invoked with plane, refIdx, startX, startY, stepX, stepY, w, h, candRow, candCol as inputs and the output is assigned to the 2D array preds[refList]. AV1 Specification at pp. 258 of 669. 14. If isCompound is equal to 1, then the variable refList is set equal to 1 and steps 5 to 13 are repeated to form the prediction for the second reference. AV1 Specification at pp. 259 of 669.

EXHIBIT 11
United States Patent No. 11,805,267
Claim Chart for Infringement of Claim19 by ASUS Accused Products

<table>
<tr><th>U.S. Patent No. 11,805,267</th><th>Accused Products</th></tr>
<tr><td></td><td>
<pre>
interpFilter = InterpFilters[candRow][candCol][1]
if (w <= 4) {
    if (interpFilter == EIGHTTAP || interpFilter == EIGHTTAP_SHARP) {
        interpFilter = 4
    } else if (interpFilter == EIGHTTAP_SMOOTH) {
        interpFilter = 5
    }
}
for (r = 0; r < intermediateHeight; r++) {
    for (c = 0; c < w; c++) {
        s = 0
        p = x + xStep * c
        for (t = 0; t < 8; t++)
            s += Subpel_Filters[interpFilter][(p >> 6) & SUBPEL_MASK][t] *
              ref[plane] [Clip3(0, lastY, (y >> 10) + r - 3)]
                           [Clip3(0, lastX, (p >> 10) + t - 3)]
        intermediate[r][c] = Round2(s, InterRound0)
    }
}
</pre>
AV1 Specification at pp. 263 of 669.</td></tr>
</table>

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS

	<pre>interpFilter = InterpFilters[candRow][candCol][0]
	if (h <= 4) {
	    if (interpFilter == EIGHTTAP || interpFilter == EIGHTTAP_SHARP) {
	        interpFilter = 4
	    } else if (interpFilter == EIGHTTAP_SMOOTH) {
	        interpFilter = 5
	    }
	}
	for (r = 0; r < h; r++) {
	    for (c = 0; c < w; c++) {
	        s = 0
	        p = (y & 1023) + yStep * r
	        for (t = 0; t < 8; t++)
	            s += Subpel_Filters[interpFilter][(p >> 6) & SUBPEL_MASK][t] *
	              intermediate[(p >> 10) + t][c]
	        pred[r][c] = Round2(s, InterRound1)
	    }
	}</pre> AV1 Specification at pp. 263 of 669.

EXHIBIT 11

UNITED STATES PATENT NO. 11,805,267

CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS

```
Subpel_Filters[ 6 ][ 16 ][ 8 ] = {
  {
    { 0, 0, 0, 128, 0, 0, 0, 0 },
    { 0, 2, -6, 126, 8, -2, 0, 0 },
    { 0, 2, -10, 122, 18, -4, 0, 0 },
    { 0, 2, -12, 116, 28, -8, 2, 0 },
    { 0, 2, -14, 110, 38, -10, 2, 0 },
    { 0, 2, -14, 102, 48, -12, 2, 0 },
    { 0, 2, -16, 94, 58, -12, 2, 0 },
    { 0, 2, -14, 84, 66, -12, 2, 0 },
    { 0, 2, -14, 76, 76, -14, 2, 0 },
    { 0, 2, -12, 66, 84, -14, 2, 0 },
    { 0, 2, -12, 58, 94, -16, 2, 0 },
    { 0, 2, -12, 48, 102, -14, 2, 0 },
    { 0, 2, -10, 38, 110, -14, 2, 0 },
    { 0, 2, -8, 28, 116, -12, 2, 0 },
    { 0, 0, -4, 18, 122, -10, 2, 0 },
    { 0, 0, -2, 8, 126, -6, 2, 0 }
  },
  {
    { 0, 0, 0, 128, 0, 0, 0, 0 },
    { 0, 2, 28, 62, 34, 2, 0, 0 },
    { 0, 0, 26, 62, 36, 4, 0, 0 },
    { 0, 0, 22, 62, 40, 4, 0, 0 },
    { 0, 0, 20, 60, 42, 6, 0, 0 },
    { 0, 0, 18, 58, 44, 8, 0, 0 },
    { 0, 0, 16, 56, 46, 10, 0, 0 },
    { 0, -2, 16, 54, 48, 12, 0, 0 },
    { 0, -2, 14, 52, 52, 14, -2, 0 },
    { 0, 0, 12, 48, 54, 16, -2, 0 },
    { 0, 0, 10, 46, 56, 16, 0, 0 },
    { 0, 0, 8, 44, 58, 18, 0, 0 },
    { 0, 0, 6, 42, 60, 20, 0, 0 },
    { 0, 0, 4, 40, 62, 22, 0, 0 },
    { 0, 0, 4, 36, 62, 26, 0, 0 },
    { 0, 0, 2, 34, 62, 28, 2, 0 }
  },
  {
    { 0, 0, 0, 128, 0, 0, 0, 0 },
    { -2, 2, -6, 126, 8, -2, 2, 0 },
    { -2, 6, -12, 124, 16, -6, 4, -2 },
    { -2, 8, -18, 120, 26, -10, 6, -2 },
    { -4, 10, -22, 116, 38, -14, 6, -2 },
    { -4, 10, -22, 108, 48, -18, 8, -2 },
    { -4, 10, -24, 100, 60, -20, 8, -2 },
    { -4, 10, -24, 90, 70, -22, 10, -2 },
    { -4, 12, -24, 80, 80, -24, 12, -4 },
    { -2, 10, -22, 70, 90, -24, 10, -4 },
    { -2, 8, -20, 60, 100, -24, 10, -4 },
    { -2, 8, -18, 48, 108, -22, 10, -4 },
    { -2, 6, -14, 38, 116, -22, 10, -4 },
    { -2, 6, -10, 26, 120, -18, 8, -2 },
    { -2, 4, -6, 16, 124, -12, 6, -2 },
    { 0, 2, -2, 8, 126, -6, 2, -2 }
  },
  {
    { 0, 0, 0, 128, 0, 0, 0, 0 },
    { 0, 0, 0, 120, 8, 0, 0, 0 },
    { 0, 0, 0, 112, 16, 0, 0, 0 },
    { 0, 0, 0, 104, 24, 0, 0, 0 },
    { 0, 0, 0, 96, 32, 0, 0, 0 },
    { 0, 0, 0, 88, 40, 0, 0, 0 },
    { 0, 0, 0, 80, 48, 0, 0, 0 },
    { 0, 0, 0, 72, 56, 0, 0, 0 },
    { 0, 0, 0, 64, 64, 0, 0, 0 },
    { 0, 0, 0, 56, 72, 0, 0, 0 },
    { 0, 0, 0, 48, 80, 0, 0, 0 },
    { 0, 0, 0, 40, 88, 0, 0, 0 },
    { 0, 0, 0, 32, 96, 0, 0, 0 },
    { 0, 0, 0, 24, 104, 0, 0, 0 },
    { 0, 0, 0, 16, 112, 0, 0, 0 },
    { 0, 0, 0, 8, 120, 0, 0, 0 }
  },
  {
    { 0, 0, 0, 128, 0, 0, 0, 0 },
    { 0, 0, -4, 126, 8, -2, 0, 0 },
    { 0, 0, -8, 122, 18, -4, 0, 0 },
    { 0, 0, -10, 116, 28, -6, 0, 0 },
    { 0, 0, -12, 110, 38, -8, 0, 0 },
    { 0, 0, -12, 102, 48, -10, 0, 0 },
    { 0, 0, -14, 94, 58, -10, 0, 0 },
    { 0, 0, -12, 84, 66, -10, 0, 0 },
    { 0, 0, -12, 76, 76, -12, 0, 0 },
    { 0, 0, -10, 66, 84, -12, 0, 0 },
    { 0, 0, -10, 58, 94, -14, 0, 0 },
    { 0, 0, -10, 48, 102, -12, 0, 0 },
    { 0, 0, -8, 38, 110, -12, 0, 0 },
    { 0, 0, -6, 28, 116, -10, 0, 0 },
    { 0, 0, -4, 18, 122, -8, 0, 0 },
    { 0, 0, -2, 8, 126, -4, 0, 0 }
  },
  {
    { 0, 0, 0, 128, 0, 0, 0, 0 },
    { 0, 0, 30, 62, 34, 2, 0, 0 },
    { 0, 0, 26, 62, 36, 4, 0, 0 },
    { 0, 0, 22, 62, 40, 4, 0, 0 },
    { 0, 0, 20, 60, 42, 6, 0, 0 },
    { 0, 0, 18, 58, 44, 8, 0, 0 },
    { 0, 0, 16, 56, 46, 10, 0, 0 },
    { 0, 0, 14, 54, 48, 12, 0, 0 },
    { 0, 0, 12, 52, 52, 12, 0, 0 },
    { 0, 0, 12, 48, 54, 14, 0, 0 },
    { 0, 0, 10, 46, 56, 16, 0, 0 },
    { 0, 0, 8, 44, 58, 18, 0, 0 },
    { 0, 0, 6, 42, 60, 20, 0, 0 },
    { 0, 0, 4, 40, 62, 22, 0, 0 },
    { 0, 0, 4, 36, 62, 26, 0, 0 },
    { 0, 0, 2, 34, 62, 30, 0, 0 }
  }
}
```

AV1 Specification at pp. 264-266 of 669.

$$\boxed{\mathrm{Round2}(x, n) = \left\lfloor \frac{x + \left(2^{n-1}\right)}{2^n} \right\rfloor}$$

AV1 Specification at pp. 19 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

```
Round2( x, n ) {
  if ( n == 0 )
    return x
  return ( x + ( 1 << (n - 1) ) ) >> n
}
```

AV1 Specification at pp. 19 of 669.

7.11.3.2. Rounding variables derivation process
The input to this process is a variable isCompound.
The rounding variables InterRound0, InterRound1, and InterPostRound are derived as follows:
• InterRound0 (representing the amount to round by after horizontal filtering) is set equal to 3.
• InterRound1 (representing the amount to round by after vertical filtering) is set equal to (isCompound ? 7 : 11).
• If BitDepth is equal to 12, InterRound0 is set equal to InterRound0 + 2.
• If BitDepth is equal to 12 and isCompound is equal to 0, InterRound1 is set equal to InterRound1 - 2.
AV1 Specification at pp. 259 of 669</td></tr>
<tr><td>[D] using said second reference block to obtain a second prediction, said second prediction having the second precision;</td><td>Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding video comprising using said second reference block to obtain a second prediction, said second prediction having a second precision, which is higher than said first precision.

For example, when isCompound is equal to 1, the steps of deriving the prediction is repeated for the second reference:

14. If isCompound is equal to 1, then the variable refList is set equal to 1 and steps 5 to 13 are repeated to form the prediction for the second reference.
AV1 Specification at pp. 259 of 669.</td></tr>
</table>

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	Accordingly, the description of AV1 prediction generation is incorporated by reference from element [C] of claim 19.
[E] obtaining a combined prediction based at least partly upon said first prediction and said second prediction;	Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding video comprising obtaining a combined prediction based at least partly upon said first prediction and said second prediction. For example, and without limitation, each of the Accused Products performs a method for decoding video comprising obtaining a combined prediction based at least partly upon said first prediction and said second prediction, corresponding to the decoding process specified by the AV1 Standard. The following specifications provide further evidence of how each of the Accused Products operates. The AV1 specification provides for 5 different compound prediction types (determined by compound_type). Each of these compound prediction types provides a combined prediction based at least partly upon said first prediction and said second prediction. If compound_type is equal to COMPOUND_AVERAGE, the AV1 specification provides the combined prediction be obtained as follows: Otherwise if compound_type is equal to COMPOUND_AVERAGE, CurrFrame[plane][y + i][x + j] is set equal to Clip1(Round2(preds[0][i][j] + preds[1][i][j], 1 + InterPostRound)) for i = 0..h-1 and j = 0..w-1. AV1 Specification at pp. 259 of 669. If compound_type is equal to COMPOUND_DISTANCE, the AV1 specification provides the combined prediction be obtained as follows: Otherwise if compound_type is equal to COMPOUND_DISTANCE, CurrFrame[plane][y + i][x + j] is set equal to Clip1(Round2(FwdWeight * preds[0][i][j] + BckWeight * preds[1][i][j], 4 + InterPostRound)) for i = 0..h-1 and j = 0..w-1. AV1 Specification at pp. 259 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	The FwdWeight and BckWeight are determined according to section 7.11.3.15. The three other compound_types use a Mask array: An array named Mask is prepared as follows: • If compound_type is equal to COMPOUND_WEDGE and plane is equal to 0, the wedge mask process in section 7.11.3.11 is invoked with w, h as inputs. • Otherwise if compound_type is equal to COMPOUND_INTRA, the intra mode variant mask process in section 7.11.3.13 is invoked with w, h as inputs. • Otherwise if compound_type is equal to COMPOUND_DIFFWTD and plane is equal to 0, the difference weight mask process in section 7.11.3.12 is invoked with preds, w, h as inputs. • Otherwise, no mask array is needed. . . . The inter predicted samples are then derived as follows: • If isCompound is equal to 0 and IsInterIntra is equal to 0, CurrFrame[plane][y + i][x + j] is set equal to Clip1(preds[0][i][j]) for i = 0..h-1 and j = 0..w-1. • Otherwise if compound_type is equal to COMPOUND_AVERAGE, CurrFrame[plane][y + i][x + j] is set equal to Clip1(Round2(preds[0][i][j] + preds[1][i][j], 1 + InterPostRound)) for i = 0..h-1 and j = 0..w-1. • Otherwise if compound_type is equal to COMPOUND_DISTANCE, CurrFrame[plane][y + i][x + j] is set equal to Clip1(Round2(FwdWeight * preds[0][i][j] + BckWeight * preds[1][i][j], 4 + InterPostRound)) for i = 0..h-1 and j = 0..w-1. • Otherwise, the mask blend process in section 7.11.3.14 is invoked with preds, plane, x, y, w, h as inputs. AV1 Specification at pp. 259 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS

	The AV1 specification describes how the above masks are applied to the predictions obtained from the first and second reference blocks (pred0 and pred1) to obtain a combined prediction:
	`for ( y = 0; y < h; y++ ) {`
	`    for ( x = 0; x < w; x++ ) {`
	`        if ( (!subX && !subY) ||`
	`             (interintra && !wedge_interintra) ) {`
	`            m = Mask[ y ][ x ]`
	`        } else if ( subX && !subY ) {`
	`            m = Round2( Mask[ y ][ 2*x ] + Mask[ y ][ 2*x+1 ], 1 )`
	`        } else if ( !subX && subY ) {`
	`            m = Round2( Mask[ 2*y ][ x ] + Mask[ 2*y+1 ][ x ], 1 )`
	`        } else {`
	`            m = Round2( Mask[ 2*y ][ 2*x ] + Mask[ 2*y ][ 2*x+1 ] +`
	`                Mask[ 2*y+1 ][ 2*x ] + Mask[ 2*y+1 ][ 2*x+1 ], 2 )`
	`        }`
	`        if ( interintra ) {`
	`            pred0 = Clip1( Round2( preds[ 0 ][ y ][ x ], InterPostRound ) )`
	`            pred1 = CurrFrame[plane][y+dstY][x+dstX]`
	`            CurrFrame[plane][y+dstY][x+dstX] = Round2( m * pred1 + (64 - m) * pred0, 6 )`
	`        } else {`
	`            pred0 = preds[ 0 ][ y ][ x ]`
	`            pred1 = preds[ 1 ][ y ][ x ]`
	`            CurrFrame[plane][y+dstY][x+dstX] = Clip1( Round2( m * pred0 + (64 - m) * pred1, 6 +`
	`InterPostRound ) )`
	`        }`
	`    }`
	`}`
	AV1 Specification at pp. 287 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS

[F] decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and	Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding video comprising decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right.
	
	For example, and without limitation, of the Accused Products performs a method for decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right. The following specifications provide further evidence of how each of the Accused Products operates:
	
	The AV1 specification provides for 5 different compound prediction types (determined by compound_type). Each of these compound prediction types leads to the application of the Round2 function with an argument based on the InterPostRound quantity.
	
	$\text{Round2}(x, n) = \left\lfloor \frac{x + (2^{n-1})}{2^n} \right\rfloor$
	AV1 Specification at pp. 19 of 669.
	
	`Round2( x, n ) {`
	`  if ( n == 0 )`
	`    return x`
	`  return ( x + ( 1 << (n - 1) ) ) >> n`
	`}`
	AV1 Specification at pp. 19 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	7.11.3.2. Rounding variables derivation process The input to this process is a variable isCompound. The rounding variables InterRound0, InterRound1, and InterPostRound are derived as follows: • InterRound0 (representing the amount to round by after horizontal filtering) is set equal to 3. • InterRound1 (representing the amount to round by after vertical filtering) is set equal to (isCompound ? 7 : 11). • If BitDepth is equal to 12, InterRound0 is set equal to InterRound0 + 2. • If BitDepth is equal to 12 and isCompound is equal to 0, InterRound1 is set equal to InterRound1 - 2. AV1 Specification at pp. 259 of 669 • InterPostRound (representing the amount to round by at the end of the prediction process) is set equal to 2 * FILTER_BITS - (InterRound0 + InterRound1). AV1 Specification at pp. 260 of 669 FILTER_BITS is 7 (see AV1 Specification at p. 14 of 669). When isCompound is 1, InterPostRound is 4 (2*7 – (3 + 7)). If compound_type is equal to COMPOUND_AVERAGE, the AV1 specification provides the combined prediction be obtained as follows, where Round2 is applied with 1 + InterPostRound as its second argument. The additional "1" added to the second argument results in an average of preds[0] and preds[1] reduced in precision to the input bit depth: Otherwise if compound_type is equal to COMPOUND_AVERAGE, CurrFrame[plane][y + i][x + j] is set equal to Clip1(Round2(preds[0][i][j] + preds[1][i][j], 1 + InterPostRound)) for i = 0..h-1 and j = 0..w-1. AV1 Specification at pp. 259 of 669. If compound_type is equal to COMPOUND_DISTANCE, the AV1 specification provides the combined prediction be obtained as follows, where Round2 is applied with 4 + InterPostRound as its second argument.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS

	The additional "4" added to the second argument reduces the bit depth added by FwdWeight and BckWeight, which add to 16 (2^4) and therefore add a total of 4 bits to the bit depth of the sum in the first argument. See AV1 Specification at p. 286 of 669. The output of the Round2 function is therefore reduced in precision to the input bit depth.
	
	Otherwise if compound_type is equal to COMPOUND_DISTANCE, CurrFrame[plane][y + i][x + j] is set equal to Clip1(Round2(FwdWeight * preds[0][i][j] + BckWeight * preds[1][i][j], 4 + InterPostRound)) for i = 0..h-1 and j = 0..w-1.
	AV1 Specification at pp. 259 of 669.
	
	The masks used by the remaining compound_types are ultimately used to determine "m" value to use in a weighted sum of the first and second prediction (pred0 and pred1) which sum to 64 (2^6), which is then passed as the first argument to Round2 with 6 + InterPostRound as the second argument. The output of the Round2 function is therefore reduced in precision to the input bit depth.
	
	`pred0 = preds[ 0 ][ y ][ x ]`
	`pred1 = preds[ 1 ][ y ][ x ]`
	`CurrFrame[plane][y+dstY][x+dstX] = Clip1( Round2( m * pred0 + (64 - m) * pred1, 6 + InterPostRound ) )`
	AV1 Specification at pp. 287 of 669.

[G] reconstructing the block of pixels based on the combined precision.	Each of the Accused Products, such as the ASUS Q543MV, performs a method for decoding video comprising reconstructing the block of pixels based on the combined precision. For example, each of the Accused Products performs a method for decoding video comprising reconstructing the block of pixels based on the combined precision, corresponding to the decoding process specified by the AV1 Specification. The following specifications provide further evidence of how each of the Accused Products operates.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS
	For example, the AV1 Specification defines a Reconstruct process in section 7.12.3 that concludes with the CurrFrame array values being added to Residual array values obtained from the inverse transform block process to yield a new CurrFrame array of reconstructed samples. **7.12.3. Reconstruct process** The reconstruct process is invoked to perform dequantization, inverse transform and reconstruction. This process is triggered at a point defined by a function call to reconstruct in the transform block syntax table described in section 5.11.35. The inputs to this process are: • a variable plane specifying which plane is being reconstructed, • variables x and y specifying the location of the top left sample in the CurrFrame[plane] array of the current transform block, • a variable txSz, specifying the size of the transform block. . . . 2. Invoke the 2D inverse transform block process defined in section 7.13.3 with the variable txSz as input. The inverse transform outputs are stored in the Residual buffer. 3. For i = 0..(h-1), for j = 0..(w-1), the following applies: ◦ The variable xx is set equal to flipLR ? (w - j - 1) : j. ◦ The variable yy is set equal to flipUD ? (h - i - 1) : i. ◦ CurrFrame[plane][y + yy][x + xx] is set equal to Clip1(CurrFrame[plane][y + yy][x + xx] + Residual[i][j]). AV1 Specification at pp. 294-295 of 669.

EXHIBIT 11
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM19 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	ACCUSED PRODUCTS